BOUNDARY ELEMENTS IN POTENTIAL AND ELASTICITY THEORY

E. ALARCÓN A. MARTÍN and F. PARÍS
Department of Structures E.T.S. Ingenieros Industriales, Polithecnical University, Madrid, Spain

Abstract—A general theory that describes the B.I.E. linear approximation in potential and elasticity problems, is developed. A method to tread the Dirichlet condition in sharp vertex is presented. Though the study is developed for linear elements, its extension to higher order interpolation is straightforward.

A new direct assembling procedure of the global of equations to be solved, is finally showed.

1. INTRODUCTION

Since the publication of Galileo's "Dialogue of two new sciences", which established the basis for the scientific approach to building problems there has been no landmark as the creation of electronic digital computer. Effectively, the creation of the "Ecole des Ponts et Chaussées" and consequently the appearance of was no more than the developement of the first "engineers" positive ideas contained in Galileo's book. The erection of the larger, better and cheaper works demanded by the nineteenth century industrial revolution, showed clearly the advantages of a scientific viewpoint over traditional empirical methods.

On its part, the computer has removed the traditional human limitations, and its impact has produced a stimulating reexamination of established knowledge in every field of science.

In structural analysis the well known 1956 paper of Turner *et al.*[1] opened one as striking way, that several other interesting numerical methods nearly fell forgotten under the splendid F.E.M. boom.

The simplicity of the method, perfectly suited to computer, attracted many researchers whose work has converted F.E.M. in an everyday tool of engineering profession.

Other methods were circumvected and only the unsurmountable difficulties for handling some problems with F.E.M. have produced the recent interest in several alternatives to it.

One of these possbilities is the Boundary Integral Equation Method (B.I.E.M.). The first idea was presented as early as 1963 by Jaswon[2] in a paper over potential problems; Rizzo[3] successfully extended the method to more complicated problems and Cruse *et al.* have shown in several papers[4–9] how the new method can cover the "misprints" of F.E.M. Several other authors have computer refinements[10], coupling B.I.E.M. with F.E.M.[11], etc. Also the first attempt to produce a book on the topic has occured[12] and two International Conferences specifically dedicated to B.I.E.M.[13, 14] have been held.

Nevertheless the mastering of the method is far from being reached and further knowledge of computer and numerical techniques are still needed to make B.I.E.M. as powerful a tool as F.E.M. is. This paper tries to be a little contribution in that way.

The implementation of B.I.E.M. offers the same difficulty when the isoparametric element concept is applied to boundary domains with sharp corners.

Standard procedures[10, 15] produce important errors if the size of the elements near the vertex is not small enough. Some authors[15] have proposed an alternative procedure in which they define in the corner as many nodes as elements are interconnected.

The brawback of both lines of action is clear; the first one multiplies the number of elements, losing one of the more outstanding features of the isoparametric representation. The latter increases the number of equations to be solved, which by the special character of B.I.E.M. is highly undesirable. Then this is the problem area we shall address.

2. B.I.E.M. TECHNIQUE IN POTENTIAL THEORY

As it is well known, the B.I.E.M. is based upon the existence of two items: a fundamental solution of the field equation and a reciprocity theorem which, in potential theory, is the Green formula[16].

Given two functions ψ and ϕ with C^2-continuity in a bounded region D and C^1 in $D+\partial D$, where ∂D_e is a piecewise smooth boundary, Green formula is

$$\int_D (\psi\nabla^2\phi - \phi\nabla^2\psi)\,\mathrm{d}v = \int_{\partial D}\left(\psi\frac{\partial\phi}{\partial\gamma} - \phi\frac{\partial\psi}{\partial\nu}\right)\mathrm{d}s. \quad (1)$$

If

$$-\nabla^2\phi = 4\pi\delta(p\cdot q)$$
$$\nabla^2\psi = 0 \quad (2)$$

being p, q two points in D, and $\delta(p, q)$ the Dirac function, one obtaines the representation

$$\psi(\underline{P}) = \frac{1}{4\pi}\int_{\partial D}\frac{1}{r}\frac{\partial\psi}{\partial\nu}\mathrm{d}s - \frac{1}{4\pi}\int_{\partial D}\psi\frac{\partial}{\partial\nu}\left(\frac{1}{r}\right)\mathrm{d}s. \quad (3)$$

For two dimensions the fundamental solution to be used is $1/2\pi\ln(1/r)$.

The particularisation of eqn (3) for points in the boundary leads to (see Ref. [16], vol. II, pp. 256–257)

$$257)c\psi(\underline{P}) + \int_{\partial D}\psi\frac{\mathrm{d}\Delta}{\mathrm{d}\nu}\mathrm{d}s = \int_{\partial D}\Delta\frac{\mathrm{d}\psi}{\mathrm{d}\nu}\mathrm{d}s \quad (4)$$

where in 3D

$$\Delta = \frac{1}{r} \quad c = \begin{cases} 4\pi & \text{for } \underline{P} \text{ in } D \\ 2\pi & \text{for } \underline{P} \text{ in } D \\ 0 & \text{for } \underline{P} \text{ outside } D. \end{cases}$$

or in 2D

$$\Delta = \ln\frac{1}{r} \quad c = \begin{cases} 2\pi & \text{for } \underline{P} \text{ in } D \\ \pi & \text{for } \underline{P} \text{ in } D \\ 0 & \text{for } \underline{P} \text{ outside } D. \end{cases}$$

if there is a continuous tangent at $\underline{P}$. If there is a conical vertex, C is the corresponding angle.

Equation (23) is now discretized following the isoparametric F.E. idea of Irons.

Both variables $(\psi, \partial\psi/\partial\nu)$ are supposed to vary in a chosen form between "nodes" in the boundary, which is also modelled with the same shape functions.

Doing

$$\psi = [N_1(\xi), N_2(\xi), N_3(\xi)\ldots]\begin{Bmatrix}\psi_1\\ \psi_2\\ \vdots\end{Bmatrix}^e$$

$$q = \partial\psi/\partial\nu = [N_1(\xi), N_2(\xi), N_3(\xi)\ldots]\begin{Bmatrix}q_1\\ q_2\\ \vdots\end{Bmatrix}^e$$

$$ds = J\,d\xi. \tag{5}$$

N_i are shape functions, ξ natural coordinates in the element and J the Jacobian to pass from the S system to the ξ one.

Equation (4) is discretized to

$$c_i\psi_i + \sum_{j=1}^{N}\left[\int_{\partial D_j} N_1\frac{\partial\Delta}{\partial\nu}J\,d\xi;\ \int_{\partial D_j} N_2\frac{\partial\Delta}{\partial\nu}J\,d\xi;\ldots\right]$$

$$\underline{\psi}^e = \sum_{j=1}^{N}\left[\int_{\partial D_j} N_2\Delta J\,d\xi;\ \int_{\partial D_j} N_2\Delta J\,d\xi;\ldots\right]\underline{q}^e. \tag{6}$$

Subindex $\underline{i}$ indicates a boundary node, and j a boundary element. The integrations are extended to every elemental subdomain ∂D_j.

The calculation of C_i is very easy. Assuming a state of constant ψ, $\underline{q}^e = 0$ and

$$c_i = -\sum_{j=1}^{N}\left[\int N_1\frac{\partial\Delta}{\partial\nu}J\,d\xi + \int N_2\frac{\partial\Delta}{\partial\nu}J\,d\xi + \cdots\right]. \tag{7}$$

Repeating eqn (6) for every node a set of algebraic equations of the type

$$\underline{H}\underline{\psi} = \underline{G}\underline{q} \tag{8}$$

can be established, where $\underline{H}$ and $\underline{G}$ are known calculated matrices and ψ, q are vectors defining the field in the boundary. Nevertheless the size of $\underline{H}$ and $\underline{G}$ is not the same.

To fix ideas, suppose a plane problem with linear interpolation. There are two shape functions:

$$N_1 = -\frac{1}{2}(\xi - 1)$$

$$N_2 = \frac{1}{2}(\xi + 1) \quad -1 < \xi < 1$$

$$ds_j = \frac{L_j}{2}d\xi \tag{9}$$

and eqn (6) is

$$2c_i\psi_i + \sum_{j=1}^{N} L_j\left[\int_{\partial D_j} N_1\frac{d}{d\nu}\left(\ln\frac{1}{r}\right)d\xi;\ \int_{\partial D_j} N_2\frac{d}{d\nu}\left(\ln\frac{1}{r}\right)d\xi\right]\begin{Bmatrix}\psi_j^j\\ \psi_{j+1}^j\end{Bmatrix} =$$

$$= \sum_{j=1}^{N} L_j\left[\int_{\partial D_j} N_1\ln\frac{1}{r}d\xi;\ \int_{\partial D_j} N_2\ln\frac{1}{r}d\xi\right]\begin{Bmatrix}q_j^j\\ q_{j+1}^j\end{Bmatrix}$$

if element j runs from j to $j+1$ node. Noting that $q_j^j \neq q_j^{j+1}$ (subindex ≡ node, superindex ≡ element) but $\psi_j^j = \psi_j^{j-1}$ it's clear that when element j is viewed from node i, there are four contributions to matrices $\underline{H}$, G. The first two are produced by the two node potentials ψ_j; ψ_{j+1}; the other is generated by q_j^j of node j and q_{j+1}^j. $\underline{H}$ is then a $N \times N$ matrix and $\underline{G}$ a $N \times 2N$, which formation scheme is as follows:

$$\begin{array}{c} \\ i \end{array}\begin{bmatrix} \cdots & \boxed{j} & \boxed{j+1} & \cdots \\ & \vdots & \vdots & \\ & \bigcirc & \bigcirc & \\ & \vdots & \vdots & \end{bmatrix}\begin{bmatrix}\vdots\\ \psi_j\\ \psi_{j+1}\\ \vdots\end{bmatrix} =$$

$$(N \times N) \qquad (N \times 1)$$

$$= \begin{bmatrix} \cdots & \boxed{2j} & \boxed{2j+1} & \cdots \\ & \vdots & \vdots & \\ & \bigcirc & \bigcirc & \\ & \vdots & \vdots & \end{bmatrix}\begin{bmatrix}\vdots\\ q_j^j\\ q_{j+1}^j\\ \vdots\end{bmatrix}. \tag{11}$$

$$(N \times 2N) \qquad (2N \times 1)$$

This fact has not been recognized in previous work [15] because in a Dirichlet condition round a node, one has two unknowns q_j^{j-1}, q_j^j and only one datum ψ_j. There are not enough equations to solve the problem. The alternative is to enforce $q_j^j = q_{j+1}^j$, reducing $\underline{G}$ to a $N \times N$ matrix but the results are very bad if there is a vertex and, moreover, in the case of a mixed condition one never knows what data to choose. In Ref. [15] this difficulty was circumvected by using two nodes in every sharp corner. The truncation of the corner produces better results in mixed situations but, as would be expected, the Dirichlet condition is not well managed.

Our alternative is directly working in eqn (11). If subindex B indicates "Before" a node, and A, "After" a node, there are five possible cases of data round a node

(1) q_B; q_A (2) q_B, ψ (3) ψ, q_A (4) ψ sharp corner
(5) ψ smooth corner.

This code is the same used in program Betis at the end of the paper. In the three first cases there are two data and only one unknown, so the use of eqn (11) solves the problem. In case 5 the obvious solution is putting $q_A = q_B$. Then only case 4 needs a special treatment.

As it is well known

$$q_B = \underline{\nabla}\psi \cdot \underline{\nu}_B = q\cos\alpha_B$$

$$q_A = \nabla\psi \cdot \underline{\nu}_A = q\cos\alpha_A \tag{12}$$

where q is the gradient modulus and α_B, α_A the angles between the gradients and boundary normals before and after the node.

If a gradient direction is assumed, the only remaining variable is q and we have now enough information to use eqn (11). But in a Dirichlet problem one always knows the derivatives along the boundary and then in a sharp corner it is possible to obtain the gradient direction through the two derivatives in it. In the linear hypothesis

$$\psi_x)_i = \frac{1}{\Delta_i}(y_i - y_{i+1};\ y_{i+1} - y_{i-1};\ y_{i-1} - y_i) \begin{Bmatrix} \psi_{i-1} \\ \psi_i \\ \psi_{i+1} \end{Bmatrix}$$

$$\psi_y)_i = \frac{1}{\Delta}(x_{i+1} - x_i;\ x_{i-1} - x_{i+1};\ x_i - x_{i-1}) \begin{Bmatrix} \psi_{i-1} \\ \psi_i \\ \psi_{i+1} \end{Bmatrix}$$

$$\Delta_i = (x_{i+1} - x_i)(y_{i-1} - \delta_i) - (x_{i-1} - x_i)(y_{i+1} - y_i)$$

$$\underline{n} = \frac{\underline{\nabla}\psi}{|\underline{\nabla}\psi|} = \left(\frac{\psi_{1x}}{\sqrt{(\psi_{1x}^2 + \psi_{1y}^2)}}; \frac{\psi_{1y}}{\sqrt{(\psi_x{}^2 + \psi_{1y}^2)}}\right)$$

where $(i-1)$, $(i+1)$ are the nodes before and after the i node under study.

Using $\underline{n}$ it is possible to compute α_B, α_A and after solving the system of N equations, a back substitution in (12) will show the desired values q_B, q_A.

The guess can be refined using an iterative process, but the run of several examples has shown very good results with this simple approach.

In the case of parabolic of higher order elements a tangent plane can always be guessed, the only change necessary being in eqn (13).

Another improvement of program Betis is the automatic fashion in which the system (11) is substituted by a

$$\underline{K}\underline{X} = \underline{F} \tag{14}$$

directly prepared to be acted by a standard routine solver.

To avoid the storage of matrices $\underline{H}$, $\underline{G}$ the program has been prepared to form the product of every calculated coefficient by its corresponding data and to put it in vector $\underline{F}$. If the code of the point under study indicates the presence of an unknown, this is relocated in the adequate row of $\underline{X}$ and so is its correlated coefficient in matrix $\underline{K}$. Repeating this process for every node one can directly form the (eqn 14) using storage for an $N \times N$ matrix and two $N \times 1$ vectors.

Careful programming has permitted also an optimal use of the in core memory of the Hewlett. Packard 21-MX in which the following examples have been run.

3. B.I.E.M. TECHNIQUE IN ELASTICITY THEORY

The reciprocity theorem in elasticity is the well known Betti's theorem

$$\int_{\partial D} \underline{T}^n \cdot \underline{u}^* \, ds + \int_D \underline{x} \cdot u^* \, dv = \int_{\partial D} \underline{T}^n \cdot \underline{u} \, ds + \int_D \underline{x}^* \underline{u} \, dv \tag{15}$$

which in absence of volume forces is reduced to

$$\int_{\partial D} \underline{T}^n \cdot \underline{u}^* \, ds = \int_{\partial D} \underline{T}^{n*} \cdot \underline{u} \, ds + \int_D \underline{x}^* \underline{u} \, dv. \tag{16}$$

The second feature of BIEM is the fundamental solution than can be obtained with a Kelvin solution

$$u_k^* = U_{k_i} e_i$$

$$T_k^{n*} = T_{k_i} e_i \tag{17}$$

where e_j are directions of actuation of unitarian loads. In two dimensions

$$u_{k_i} = \frac{1}{8\pi\mu(1-\nu)}\left[(3-4\nu)\ln\frac{1}{r}\delta_{k_i} + r_{1k}r_{,i}\right]$$

$$T_{k_i} = -\frac{1}{4\pi(1-\nu)}\frac{1}{r}\left\{\frac{dr}{dn}[(1-2\nu)\delta_{k_i} + 2r_{,i}r_{2k}]\right.$$

$$\left. + (1-2\nu)[n_i r_{,k} - n_k r_{,i}]\right\}. \tag{18}$$

Where $\underline{T}$ are the tractions, $\underline{u}$ the displacement vector, r the distance between the point under study and that in which the unit load is placed, $\mu = E/2(1+\nu)$, E. Young modulus and ν Poisson coefficient.

Substituting the Kelvin solution in (16) one gets

$$u_i(p) + \int_{\partial D} T_{k_i}(p,q) u_k(q) \, ds = \int_{\partial D} u_{k_i}(p,q) T_k^n(q) \, ds \tag{19}$$

which is known as Somigliana's identity.

Up from here the process is absolutely the same used in the previous paragraph.

Establishing the equation in the contour,

$$c(\underline{p})\underline{u}(\underline{p}) + \int_{\partial D} \underline{T} \cdot \underline{u} \, ds = \int_{\partial D} \underline{U} \cdot \underline{T}^n \, ds. \tag{20}$$

Doing the linear discretization

$$\underline{u} = \begin{Bmatrix} u \\ v \end{Bmatrix} = \left[\begin{array}{cc|cc} N_1 & N_2 & 0 & 0 \\ 0 & 0 & N_1 & N_2 \end{array}\right] \begin{Bmatrix} \bar{X}_1^e \\ \bar{X}_2^e \\ \bar{Y}_1^e \\ \bar{Y}_2^e \end{Bmatrix}$$

$$ds = \frac{L^e}{2} d\xi$$

$$N_1 = -\frac{1}{2}(\xi - 1)$$

$$N_2 = +\frac{1}{2}(\xi + 1) \tag{21}$$

with the usual notations, eqn (20) can be written for every node i

$$2c_i \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} + \sum_{j=1}^{N} L_j \left[\begin{array}{cc|cc} \int_{\partial D_j} N_1 T_{11}\, d\xi & \int_{\partial D_j} N_2 T_{11}\, d\xi & \int_{\partial D_j} N_1 T_{12}\, d\xi & \int_{\partial D_j} N_2 T_{12}\, d\xi \\ \int_{\partial D_j} N_1 T_{21}\, d\xi & \int_{\partial D_j} N_2 T_{21}\, d\xi & \int_{\partial D_j} N_1 T_{22}\, d\xi & \int_{\partial D_j} N_2 T_{22}\, d\xi \end{array}\right] \begin{Bmatrix} u_j^j \\ u_{j+1}^j \\ v_j^j \\ v_{j+1}^j \end{Bmatrix} =$$

$$= \sum_{j=1}^{N} L_j \left[\begin{array}{cc|cc} \int_{\partial D_j} N_1 U_{11}\, d\xi & \int_{\partial D_j} N_2 U_{11}\, d\xi & \int_{\partial D_j} N_1 U_{12}\, d\xi & \int_{\partial D_j} N_2 U_{12}\, d\xi \\ \int_{\partial D_j} N_1 U_{11}\, d\xi & \int_{\partial D_j} N_2 U_{21}\, d\xi & \int_{\partial D_j} N_1 U_{22}\, d\xi & \int_{\partial D_j} N_2 U_{22}\, d\xi \end{array}\right] \begin{Bmatrix} \bar{X}_j^j \\ \bar{X}_{j+1}^j \\ \bar{Y}_j^j \\ \bar{Y}_{j+1}^j \end{Bmatrix}. \tag{22}$$

As usual a subindex means "node" and a superindex "element". $(\bar{X}, \bar{Y})$ are the coordinates components of boundary tractions and T_{kl}, U_{ke} are the components of the matrices $\underline{T}$ and $\underline{U}$ of the fundamental solution used.

In this point a transformation will be made: displacements will remain in global coordinates but tractions will be expressed as normal an tangential components.

$$\begin{Bmatrix}\sigma\\ \tau\end{Bmatrix}^e = \begin{pmatrix}\cos\alpha_e & \sin\alpha_e\\ -\sin\alpha_e & \cos\alpha_e\end{pmatrix}\begin{Bmatrix}\bar{X}\\ \bar{Y}\end{Bmatrix} \tag{23}$$

where α_e is the x-axis, normal to element) angle.

Equation (22) will now be

$$c_i\begin{Bmatrix}u_i\\ v_i\end{Bmatrix} + \sum_{j=1}^{N}\begin{bmatrix}A^j_{11} & A^j_{12} & A^j_{13} & A^j_{14}\\ A^j_{21} & A^j_{22} & A^j_{23} & A^j_{24}\end{bmatrix}\begin{Bmatrix}u^j_j\\ u^j_{j+1}\\ v^j_j\\ v^j_{j+1}\end{Bmatrix} = \sum_{j=1}^{N}\begin{bmatrix}B^j_{11} & B^j_{12} & B^j_{13} & B^j_{14}\\ B^j_{21} & B^j_{22} & B^j_{23} & B^j_{24}\end{bmatrix}\begin{Bmatrix}\sigma^j_j\\ \sigma^j_{j+1}\\ \tau^j_j\\ \tau^j_{j+1}\end{Bmatrix}. \tag{24}$$

Varying i one can obtain a system of $2N$ equations

$$\underline{A}\,\underline{U} = \underline{B}\underline{\sigma}. \tag{25}$$

The scheme of contribution of each element is f or matrix H

$$\begin{array}{c} \\ (i) \\ (N+i) \end{array}\begin{bmatrix} \overset{(j)}{\vdots} & \overset{(j+1)}{\vdots} & & \overset{(N+j)}{\vdots} & \overset{(N+j+1)}{\vdots} \\ \cdots A^j_{11} & A^j_{12} & \cdots & A^j_{13} & A^j_{14}\cdots \\ \vdots & \vdots & & \vdots & \vdots \\ \cdots A^j_{21} & A^j_{22} & \cdots & A^j_{23} & A^j_{24}\cdots \\ \vdots & \vdots & & \vdots & \vdots \end{bmatrix}\begin{Bmatrix}\vdots\\ u_j\\ u_{j+1}\\ \vdots\\ v_j\\ v_{j+1}\\ \vdots\end{Bmatrix} = \quad (2N+1)$$

$$(2N\times 2N)$$

$$= \begin{array}{c} \\ (i) \\ (N+i) \end{array}\begin{bmatrix} \overset{(2j)}{\vdots} & \overset{(2j+1)}{\vdots} & & \overset{(2N+2j)}{\vdots} & \overset{(2N+2j+1)}{\vdots} \\ \cdots B^j_{11} & B^j_{12} & \cdots & B^j_{13} & B^j_{14}\cdots \\ \vdots & \vdots & & \vdots & \vdots \\ \cdots B^j_{21} & B^j_{22} & \cdots & B^j_{23} & B^j_{24}\cdots \\ \vdots & \vdots & & \vdots & \vdots \end{bmatrix}\begin{Bmatrix}\vdots\\ \sigma^j_j\\ \sigma^j_{j+1}\\ \vdots\\ \tau^j_j\\ \tau^j_{j+1}\\ \vdots\end{Bmatrix}$$

As in the previous paragraph some authors reduce the size of B equating the tractions round each node[10]. Nevertheless one has enough data use eqn (25) except when the conditions are in displacements, because then there are four unknowns: σ_B, τ_B, σ_A, τ_A and only two equations to be used.

The reduction of 4 unknowns to two follows the same pattern as in potential theory, *the dummy variables being now the principal stresses in the corner.*

If one defines

$$\begin{aligned}\cos\vartheta &= \underline{n}_{\mathrm{I}}\cdot\underline{n}\\ \sin\vartheta &= \underline{n}_{\mathrm{II}}\cdot\underline{n}\end{aligned} \tag{26}$$

where n is the element normal and $\underline{n}_{\mathrm{I}}$, $\underline{n}_{\mathrm{II}}$ the principal directions of stresses, it can be easily shown

$$\begin{Bmatrix}\sigma\\ \tau\end{Bmatrix}^n = \left[\begin{array}{c|c}\cos^2\alpha & \sin^2\alpha\\ \hline -\sin\alpha\cos\alpha & \sin\alpha\cos\alpha\end{array}\right]\begin{Bmatrix}\sigma_{\mathrm{I}}\\ \sigma_{\mathrm{II}}\end{Bmatrix} \tag{27}$$

being σ_{I}, σ_{II} the principal stresses.

Transformation(27) is followed in corners where only displacements are known.

To obtain n_{I} and n_{II}, u and v, are assumed to behave as plane functions near the corner. Then one can define

$$\begin{aligned}\epsilon_x &= u_{,x}\\ \epsilon_y &= v_{,y}\\ \epsilon_{xy} &= u_{,y} + v_{,x}\end{aligned} \tag{28}$$

in function of the $(u_{i-1}; u_i; u_{i+1})$, $(v_{i-1}; v_i; v_{i+1})$ values in nodes $(i-1)$; $(i+1)$ before and after that in study.

Eigenvalues of $\underline{\epsilon}$ tensor

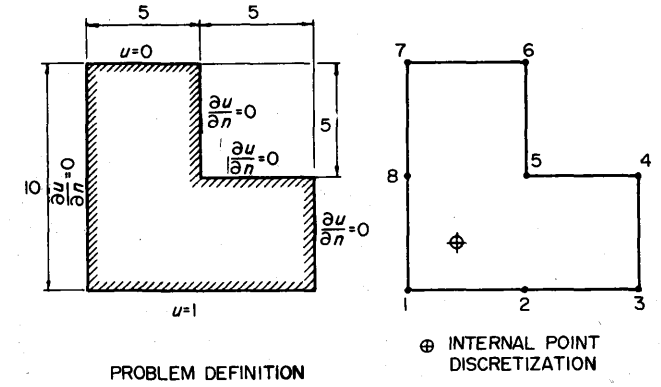
$$\epsilon_{\mathrm{I,II}} = \frac{\epsilon_x+\epsilon_y}{2} \pm \frac{1}{2}\sqrt{((\epsilon_x-\epsilon_y)^2 + 4\gamma^2_{xy})} \tag{29}$$

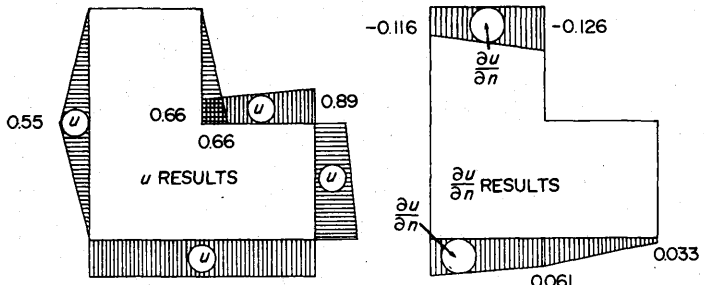
and eigenvectors

$$\underline{n}_{\mathrm{I,II}} = \frac{1}{\sqrt{((\epsilon_y-\epsilon)^2+\gamma^2_{xy})}}\begin{Bmatrix}\epsilon_y-\epsilon\\ -\gamma_{xy}\end{Bmatrix} \tag{30}$$

will be subsequently obtained.

Equation (30) are the normals needed in eqn (26) because in isotropic elasticity principal axis are common to stress and strain tensors.

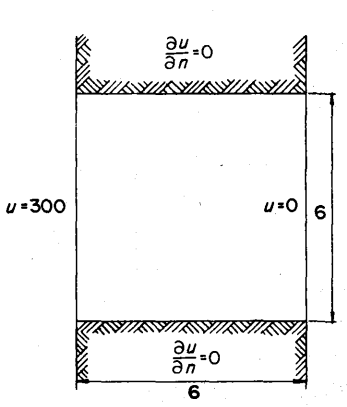


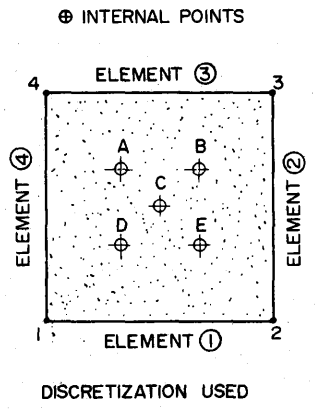


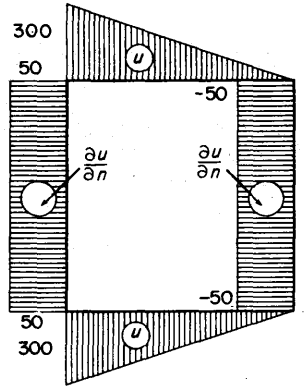
Node	Present Method	Reference†	Reference‡	Internal Point
1	u=1.00	u=1.00	u=1.00	Pres. Method
2	1.00	1.00	1.00	0.8360
3	1.00	1.00	1.00	
4	0.89389	0.9005	0.9009	Reference
5	0.66747	0.6663	0.6667	0.8388
6	0.00	0.00	0.00	
7	0.00	0.00	0.00	Reference
8	0.55523	0.5500	0.5495	0.8387

† Papamichael ‡ Jaswon

Fig. 1.







BOUNDARY NODES

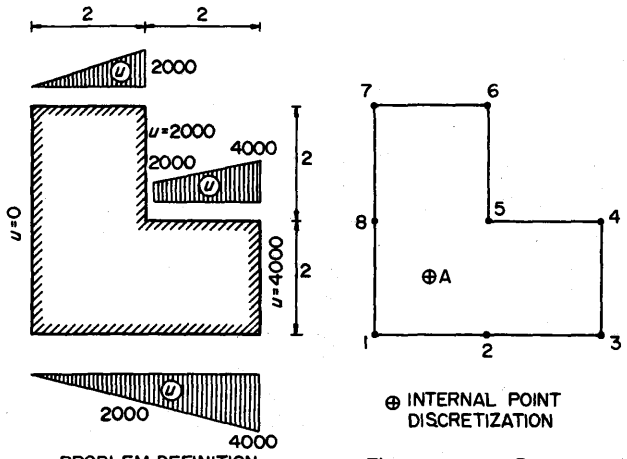
Node	Theory	Present method	Reference a	b	c
1	300	300.00000	252.2	300.0	298.0
2	-50	-50.00037	-52.9	-24.0	-50.13
3	-50	-50.00040	-52.9	-24.0	-50.13
4	300	300.00000	252.2	300.0	298.0

INTERNAL POINTS

Node	Theory	Present method	Reference a	b	c
A	200	201.376	200.3	200.4	200.0
B	100	100.014	99.7	99.7	100.0
C	150	149.793	150.0	150.0	150.0

a – Constant elements
b – Linear elements
c – Linear binode elements

Fig. 2.



Node	Theory before	Theory after	Present method before	Present method after
1	−1000	0	−999.99	0
2	0	0	−0.002	−0.002
3	0	1000	0	1000.007
4	1000	0	999.99	0
5	0	1000	0	1000.009
6	1000	0	1000.009	0
7	0	−1000	0	−1000.003
8	−1000	−1000	−1000.009	−1000.009

Internal points

	Theory	Present method
A	1000	999.720

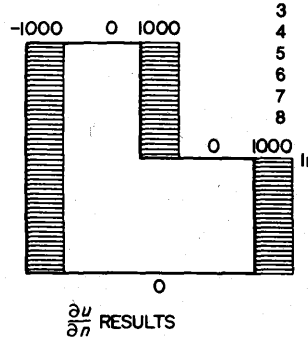
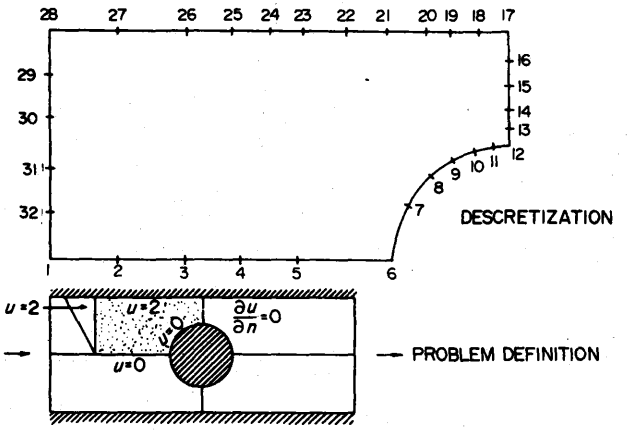
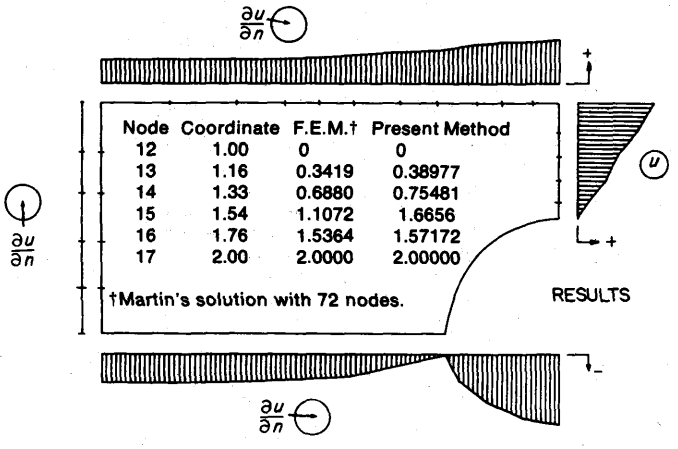


Fig. 4.

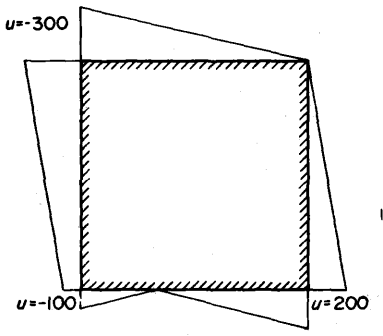




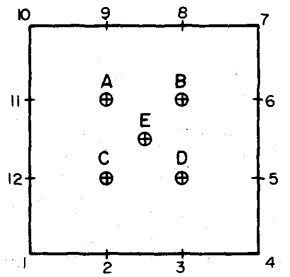
Node	Coordinate	F.E.M.†	Present Method
12	1.00	0	0
13	1.16	0.3419	0.38977
14	1.33	0.6880	0.75481
15	1.54	1.1072	1.6656
16	1.76	1.5364	1.57172
17	2.00	2.0000	2.00000

†Martin's solution with 72 nodes.

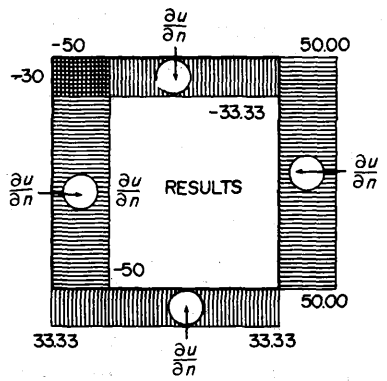
Fig. 3.



PROBLEM DEFINITION



⊕ INTERNAL POINT
DISCRETIZATION



	Theory		Present method	
Node	before	after	before	after
1	−50.00	33.33	−50.0000	33.3336
2	33.33	33.33	33.3336	33.3331
3	33.33	33.33	33.3331	33.3331
4	33.33	50.00	33.3334	50.0001
5	50.00	50.00	49.9998	49.9998
6	50.00	50.00	50.0003	50.0003
7	50.00	−33.33	−49.9996	−33.3331
8	−33.33	−33.33	−33.3336	−33.3336
9	−33.33	−33.33	−33.3331	−33.3331
10	−33.33	−50.00	−33.3333	−50.0000
11	−50.00	−50.00	−49.9999	−49.9999
12	−50.00	−50.00	−50.0001	−50.0001

Interior points

	Theory	Present method
A	−133.3333	−133.334
B	−33.3333	−33.333
C	−66.6666	−66.667
D	33.3333	33.334
F	−50.0000	−50.000

Fig. 5.

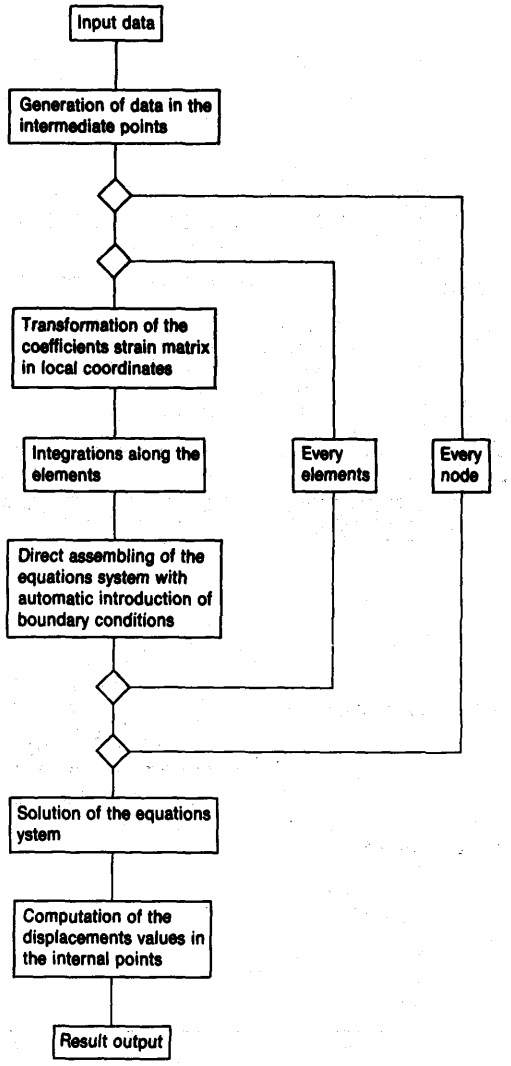


Fig. 6. Flow chart program SERBA.

As in potential theory this guess have proved to be very efficient.

The program Serba, whose flow-chart is presented in Fig. 6, uses this improvement. In it eqn (24) is not managed. Instead a system

$$\underline{K}\underline{X} = \underline{F} \tag{31}$$

is *directly formed* following the same philosophy than in potential theory.

As previously noted this direct procedure of assemblage is very well suited to small computers.

4. FUTURE TRENDS IN B.I.E. METHODS

Several interesting advantages can be atributed to B.I.E.M. over F.E. techniques.

(1) The ease of treatment of infinite domains which arises in soil-mechanics or hydraulics problems.

This is specially appreciated in dynamic soil-structure or fluid-structure problems, when F.E.M. needs a new tool: the absorbing boundary.

(2) If F.E.M. is established on displacements, the stresses are worse modelled than those. In B.I.E.M. the internal results have the same degree of accuracy. This is very interesting when one is studying situations in which concentration of stresses are present.

(3) The reduction in dimension (3D→2D: 2D→1D) of the domain to be discretized also reduces the size of the problem to be handled. This is important because it means a reduction in the input data to be prepared, the interpretation of results and the in-core computer occupation.

All this features will push the applications of B.I.E.M. to dynamics, plasticity, termoelasticity, viscoplasticity, anisotropic problems, etc. Several papers have already been published and it is clear that, in the future, more work will be dedicated to these fields.

In the other hand, in spite of its virtues, B.I.E.M. have gross problems in front of its computer implementation.

The matrices to be used are fully populated and nonsymetric. Another future trend will be the implementation of good numerical techniques to treat this situation and, also, the establishment of new procedures (substructuring, subregionalization, etc.)

Finally more effort will be dedicated to the coupling with F.E. in order to obtain the best possible analysis capability.

5. CONCLUSIONS

(i) A comprehensive treatment of potential and elasticity problems, in the light of B.I.E.M. technique, has been outlined.

(ii) Two new ideas described are the treatment of sharp-corner boundaries and the direct assembling of the equation system. Both ideas can be easily extended to all classes of isoparametric elements and dimension problems.

(iii) Two simple computer programs have been developed to show the ease implementation of the procedure.

(iv) The method has been applied to a number of problems showing its good work quality.

REFERENCES

1. M. J. Turner, R. W. Clough, H. C. Martin and L. J. Topp, Stiffness and deflection analysis of complex structures. *J. Aero. Sci.* **23**(9) (1956).
2. M. A. Jaswon, Integral equation methods in potential theory. *Proc. R. Soc.* **A275**, 23–32 (1963).
3. F. J. Rizzo, An integral equation approach to boundary value problems of classical elastostatics. *Q. Appl. Math.* **25**(1), 83–95 (1967).
4. T. A. Cruse, Mathematical Foundations of the B.I.E.M. in Solid Mechanics AFOSR F 44620-74-0060 (1977).
5. T. A. Cruse, Numerical solutions in 3-D elastostatics. *Int. J. Solids Structures* **5**, 1259–1274 (1969).
6. T. A. Cruse, Numerical evaluation of elastic stress intensity factors by the B.I.E.M. In *The Surface Crack: Physical Problems and Computational Solutions*. Swedlow, ASCE, New York (1972).
7. T. A. Cruse, Application of the B.I.E.M. to 3-d stress analysis. *Comput. Structures* **3**, 509–527 (1973).
8. T. A. Cruse, An improved B.I.E.M. for 3-d elastic-stress analysis. *Comput. Structures* **3**, 509–527 (1974).
9. T. A. Cruse, D. W. Snow and R. B. Wilson, Numerical solutions in axisymmetric elasticity. *Comput. Structues* (1977).
10. J. C. Lachat and J. O. Watson, Effective numerical treatment of B.I.E.M. *Int. J. Num. Meth. Engng* **10**, 991–1005 (1975).
11. O. C. Zienkiewickz, D. W. Kelly and P. Bettes, The compling of the F.E.M. and boundary solution procedures. *Int. J. Num. Meth. Engng* **11**, 355–375 (1977).
12. M. A. Jaswon and G. T. Symm, *Integral Equation Methods in Potential Theory and Elastostatics*. Academic Press, New York.
13. T. A. Cruse and F. J. Rizzo (Editor), *B.I.E.M.: Computational Applications in Applied Mechanics* AMD-Vol. 11. ASME, New York (1975).
14. C. Brebbia (Editor), *Recent Advances in B.E.M.* University of Southampton (1978).
15. C. Brebbia and J. Dominguez, B.E.M. for potential problems. *Appl. Math. Modelling* **2**(7) (1977).
16. R. Courant and D. Hilbert, *Methods of Mathematical Physics*. Wiley, New York (1962).
17. N. Papamichael and J. R. Whiteman, A numerical conformal transformation method for harmonic mixed boundary value problems in poligonal domains. *J. Appl. Math. Phys. (ZAMP)*, **24**, 304–316 (1973).

APPENDIX

HEAT FLUX IN A PLATE WITH FOUR ELEMENTS

NODE	COOR X	COOR Y	CODE	FUNCTION	FLUX BEFORE	FLUX AFTER
1	.0000	.0000	2	300.00000	.00000	.00000
2	6.0000	.0000	3	.00000	.00000	.00000
3	6.0000	6.0000	2	.00000	.00000	.00000
4	.0000	6.0000	3	300.00000	.00000	.00000

HEAT FLUX IN A PLATE WITH FOUR ELEMENTS

BOUNDARY POINTS

NODE	FUNCTION	FLUX BEFORE	FLUX AFTER
1	300.00000	50.00040	.00000
2	.00000	.00000	-50.00037
3	.00000	-50.00040	.00000
4	300.00000	.00000	50.00031

```
 INTERNAL POINTS
 COOR X          COOR Y          FUNCTION
 2.000           2.000           201.376
 2.000           4.000           201.376
 3.000           3.000           149.793
 4.000           2.000           100.014
 4.000           4.000           100.014


NASAS  T=00003 IS ON CR00010 USING 00010 BLKS  R=0000
001  FTN4,L
002        SUBROUTINE HGNUM(XP,YP,X1,Y1,X2,Y2,A1,A2,B1,B2)
003  C
004  C  THIS SUBROUTINE COMPUTE NUMERICALLY THE INTEGRALS ALONGS AN ELEMENT
005  C  WHICH DOES NOT INCLUDE THE NODE UNDER CONSIDERATION
006  C
007        DIMENSION XI(4),OMEG(4)
008        DATA XI/0.86113631,-0.86113631,0.33998104,-0.33998104/,
009       1   OMEG/0.34785485,0.34785485,0.65214515,0.65214515/
010        AX= (X2-X1)/2
011        AY= (Y2-Y1)/2
012        BX=(X2+X1)/2
013        BY=(Y2+Y1)/2
014        IF(AX)10,20,10
015     10 DIS=ABS((AY*XP/AX-YP+Y1-AY*X1/AX)/SQRT((AY/AX)**2+1))
016        GO TO 30
017     20 DIS=ABS(XP-X1)
018     30 SIG=(X1-XP)*(Y2-YP)-(X2-XP)*(Y1-YP)
019        IF(SIG.LT.0) DIS=-DIS
020        A1=0.
021        A2=0.
022        B1=0.
023        B2=0.
024        DO 40 I=1,4
025        XC=AX*XI(I)+BX
026        YC=AY*XI(I)+BY
027        R=SQRT((XP-XC)**2+(YP-YC)**2)
028        H=DIS*OMEG(I)*SQRT(AX**2+AY**2)/R**2
029        G=ALOG(1/R)*OMEG(I)*SQRT(AX**2+AY**2)
030        A1=A1+(XI(I)-1)*H/2
031        A2=A2-(XI(I)+1)*H/2
032        B1=B1-(XI(I)-1)*G/2
033     40 B2=B2+(XI(I)+1)*G/2
034        RETURN
035        END
036        SUBROUTINE HGANA(X1,Y1,X2,Y2,A1,A2,B1,B2)
037  C
038  C  THIS SUBROUTINE COMPUTE ANALITICALLY THE INTEGRALS ALONG AN ELEMENT
039  C  INCLUDING THE NODE UNDER CONSIDERATION
040  C
041        A1=0.
042        A2=0.
043        DIS=SQRT((X2-X1)**2+(Y2-Y1)**2)
044        B1=DIS*(1.5-ALOG(DIS))/2
045        B2=DIS*(0.5-ALOG(DIS))/2
046        RETURN
047        END
048        SUBROUTINE SOLUG(A,N,C,IPIVO,NPMAX)
049  C
050  C  THIS SUBROUTINE SOLVE LINEAR SYSTEMS OF EQUATIONS BY THE GAUSS
051  C  ELIMINATION METHOD
052  C
053        DIMENSION A(NPMAX,NPMAX),IPIVO(NPMAX),C(NPMAX)
054        IMP=6
055        DO 9 M=1,N
056      9 IPIVO(M)=0
057        DO 55 J=1,N
058        DMAX=0.0
059        DO 200 K=1,N
060        IF(IPIVO(K)-1)31,200,31
061     31 IF(ABS(DMAX)-ABS(A(K,K)))32,200,200
062     32 DMAX=A(K,K)
063        IR=K
064    200 CONTINUE
065        IF(ABS(DMAX)-2.0E-8)600,600,40
066    600 WRITE(IMP,99)
067     99 FORMAT("THE MATRIX IS SINGULAR")
068     40 IPIVO(IR)=IPIVO(IR)+1
069        A(IR,IR)=1.0
070        DO 41 I=1,N
071     41 A(IR,I)=A(IR,I)/DMAX
072        C(IR)=C(IR)/DMAX
073        DO 55 K=1,N
074        IF(K-IR)42,55,42
075     42 R=A(K,IR)
076        A(K,IR)=0.0
077        DO 45 L=1,N
078     45 A(K,L)=A(K,L)-A(IR,L)*R
079        C(K)=C(K)-C(IR)*R
080     55 CONTINUE
081        RETURN
082        END
083  $
```

```
NASAP  T=00003 IS ON CR00010 USING 00024 BLKS R=0000
001 FTN4,L
002       PROGRAM BETIS(3)
003 C
004 C  THIS IS A THREE SEGMENTS PROGRAM
005 C  SEGMENT BETI1: GENERATION INPUT DATA
006 C  SEGMENT BETI2: FORMULATION AND SOLUTION OF THE SYSTEM OF EQUATIONS
007 C  SEGMENT BETI3: INTERNAL POINTS COMPUTATIONS
008 C
009       DIMENSION NOM(3)
010       COMMON ACOM(2410)
011       DATA NOM/2HBE,2HTI,2H1 /
012       CALL EXEC (8,NOM)
013       END
014 C
015 C  FIRST SEGMENT STARTING
016 C
017       PROGRAM BETI1(5)
018       DIMENSION X(47),Y(47),NCOD(47),IENC(25),NOM(3),FI(47),DFI(47,2)
019       DIMENSION XINT(46),YINT(46)
020       COMMON IMP,LEC,NP,NP1,NSUP,IENC,NPMAX,X,Y,NCOD,FI,DFI,XINT,YINT
021       DATA NOM/2HBE,2HTI,2H2 /
022 C
023 C  ASSIGN DATA SET NUMBERS FOR INPUT, LEC, AND OUTPUT, IMP
024 C
025       LEC=7
026       IMP=6
027       NPMAX=46
028 C
029 C  READ BASIC PARAMETERS
030 C  NP=NUMBER OF BOUNDARY NODES
031 C  NPY=NUMBER OF INTERNAL POINTS WHERE THE FUNCTION WILL BE CALCULATED
032 C  IENC=PROBLEM TITLE
033 C
034       READ(LEC,1000) NP,NP1,IENC
035       DO 7 I=1,NP
036       FI(I)=0.
037       DFI(I,1)=0.
038     7 DFI(I,2)=0.
039  1000 FORMAT(2I5,10X,25A2)
040 C
041 C  READ BASIC PARAMETERS
042 C  XINT,YINT= INTERNAL POINTS COORDINATES
043 C
044       READ(LEC,1005)(XIN(I),YINT(I),I=1,NP1)
045  1005 FORMAT(2F10.0)
046       L=0
047 C
048 C  READ BASIC PARAMETERS
049 C  X,Y=BOUNDARY POINTS COORDINATES
050 C
051     5 READ(LEC,1010)I,X(I),Y(I)
052  1010 FORMAT(I5,2F10.0)
053       IF(L)15,10,15
054    15 NINT=I-L
055       AX=(X(I)-X(L))/NINT
056       AY=(Y(I)-Y(L))/NINT
057    10 L=L+1
058       IF(I-L)500,25,20
059    20 X(L)=X(L-1)+AX
060       Y(L)=Y(L-1)+AY
061       GO TO 10
062    25 IF(NP-I)24,26,5
063    26 X(NP+1)=X( 1)
064       Y(NP+1)=Y(1)
065    24 L=0
066 C
067 C  READ BASIC PARAMETERS
068 C  NCOD(I)=CODE OF THE NODE I ACCORDING TO THE NEIGHBOURG CONDITIONS
069 C        CODE 1 = THE VALUES OF THE FUNCTION DERIVATIVE BEFORE AND
070 C                 AFTER OF THE NODE ARE KNOWN
071 C        CODE 2 = THE VALUE OF THE FUNCTION BEFORE AND THE VALUE OF THE
072 C                 FUNCTION DERIVATIVE AFTER ARE KNOWN
073 C        CODE 3 = THE VALUE OF THE FUNCTION DERIVATIVE BEFORE AND THE
074 C                 VALUE OF THE FUNCTION AFTER ARE KNOWN
075 C        CODE 4 = THE VALUES OF THE FUNCTION BEFORE AND AFTER A SHARP
076 C                 CORNER ARE KNOWN
077 C        CODE 5 = US 4 BUT IN CONTINOUS BOUNDARY
078 C  FI(I) = THE VALUE OF THE FUNCTION IN THE NODE I, IF IS KNOWN
079 C  DFI(I,1),DFI(I,2) = THE VALUES OF THE DERIVATIVE FUNCTIONS BEFORE
080 C                      AND AFTER IF IS KNOWN
081 C
082    30 READ(LEC,1015)I,NCOD (I),FI(I),DFI (I,1),DFI(I,2)
083  1015 FORMAT(2I5,3F10.0)
084       IF(I-L-1)502,35,40
085    40 NINT=I-L
086       AX=(FI(I)-FI(L))/NINT
087       AY=(DFI(I,1)-DFI(L,2))/NINT
088    35 L=L+1
089       IF(I-L)502,55,42
090    42 IF(NCOD (I).LE.3.AND.NCOD (I).NE.2) GO TO 41
091       NCOD(L)=5
092       FI(L)=FI(L-1)+AX
```

```
093         GO TO 35
094      41 NCOD(L)=1
095         DFI(L,1)=DFI(L-1,2)+AY
096         DFI(L,2)=DFI(L,1)
097         GO TO 35
098      55 IF(NP-1)54,56,30
099      56 FI(NP+1)=FI(1)
100         DFI(NP+1,1)=DFI(1,1)
101         DFI(NP+1,2)=DFI(1,2)
102      54 WRITE(IMP,2000)IENC
103    2000 FORMAT(1H1,10X,25A2/3X,"NODE",12X,"COOR X",9X,"COOR Y",9X," CODE "
104        1,10X," FUNCTION",6X,"   FLUX BEFORE",5X,"       FLUX AFTER")
105         WRITE(IMP,1020)(I,X(I),Y(I),NCOD(I),FI(I),(DFI(I,K),K=1,2),I=1,NP)
106    1020 FORMAT(2X,I5,3X,2F15.4,I5,1X,3F20.5)
107     500 CONTINUE
108     502 CONTINUE
109         CALL EXEC (8,NOM)
110         END
111  C
112  C  SECOND SEGMENT STARTING
113  C
114         PROGRAM BET12(5)
115         DIMENSION X(47),Y(47),NCOD(47),IENC(25),NOM(3),FI(47),DFI(47,2)
116         DIMENSION XINT(46),YINT(46)
117         DIMENSION COEF(46,46),CARGA(46),A(2),B(2),IPIVO(46)
118         COMMON IMP,LEC,NP,NFI,NSUP,IENC,NPMAX,X,Y,NCOD,FI,DFI,XINT,YINT
119         DATA NOM/2HBE,2HT1,2H3 /
120         DIS(X1,Y1,X2,Y2)=SQRT((X1-X2)**2+(Y1-Y2)**2)
121         DO 100 NODO=1,NP
122         DO 10 I=1,NF
123      10 COEF(NODO,I)=0.
124         CDIAG=0.
125         CARGA(NODO)=0.
126  C
127  C  IF THE ELEMENTS INCLUDE THE NODE UNDER CONSIDERATION,THE INTEGRATION
128  C  ON THE BOUNDARY ELEMENTS ARE COMPUTED ANALITICALLY (SUBROUTINE HGANA),
129  C  IF NOT THOSE ARE NUMERICALLY DONE (SUBROUTINE HGNUM)
130  C
131         DO 90 IELEM=1,NP
132         IF(IELEM.EQ.NODO)GO TO 11
133         IF((IELEM+1).EQ.NODO)GO TO 12
134         IF(IELEM.EQ.NP.AND.NODO.EQ.1)GO TO 12
135         CALL HGNUM(X(NODO),Y(NODO),X(IELEM),Y(IELEM),X(IELEM+1),
136        1Y(IELEM+1),A(1),A(2),B(1),B(2))
137         GO TO 15
138      11 CALL HGANA(X(NODO),Y(NODO),X(IELEM+1),Y(IELEM+1),A(1),A(2),B(1),
139        1B(2))
140         GO TO 15
141      12 CALL HGANA(X(IELEM),Y(IELEM),X(NODO),Y(NODO),A(1),A(2),B(1),B(2))
142         AX=B(1)
143         B(1)=B(2)
144         B(2)=AX
145      15 DIAG=CDIAG+A(1)+A(2)
146         DO 80 K=1,2
147         KN=IELEM+K-1
148         IF(KN.EQ.NF+1)KN=1
149  C
150  C  THE INTEGRATIONS RESULTS ARE PLACED IN THE COEFFICIENTS MATRIX OR IN
151  C  THE LOAD VECTOR ACCORDING TO THE CODE OF NODES
152  C
153         GO TO(1,2,3,4,5),NCOD(KN)
154       1 CARGA(NODO)=CARGA(NODO)+B(K)*DFI(KN,3-K)
155         COEF(NODO,KN)=COEF(NODO,KN)+A(K)
156         GO TO 80
157       2 IF(K.EQ.2)GO TO 31
158      21 CARGA(NODO)=CARGA(NODO)+B(K)*DFI(KN,3-K)-A(K)*FI(KN)
159         GO TO 80
160       3 IF(K.EQ.2)GO TO 21
161      31 CARGA(NODO)=CARGA(NODO)-A(K)*FI(KN)
162         COEF(NODO,KN)=COEF(NCDO,KN)-B(K)
163         GO TO 80
164       4 DEN=DIS(X(IELEM),Y(IELEM),X(IELEM+1),Y(IELEM+1))
165         XNUE=(Y(IELEM+1)-Y(IELEM))/DEN
166         YNUE=(X(IELEM)-X(IELEM+1))/DEN
167         KNMA=KN+1
168         KNME=KN-1
169         IF(KN.EQ.1)KNME=NP
170         IF(KN.EQ.NP)KNMA=1
171         DEN=(X(KNMA)-X(KN))*(Y(KNME)-Y(KN))-(X(KNME)-X(KN))*(Y(KNMA)-Y(KN)
172        1)
173  C
174  C  COMPUTE GRADIENT COMPONENTS IN THE NODE UNDER STUDY
175  C
176         DFX=((Y(KN)-Y(KNMA))*FI(KNME)+((Y(KNMA)-Y(KNME))*FI(KN)+(Y(KNME-Y(
177        1KN))  FI(KNMA))/DEN
178         DFY=((X(KNMA)-X(KN))*FI(KNME)+(X(KNME)-X(KNMA))*FI(KN)+(X(KN)-X(KN
179        1ME))  FI(KNMA))/DEN
180         COSE=0.7071
181         IF(DFX.NE.0.) GO TO 43
182         IF(DFY.EQ.0.) GO TO 44
183      43 COSE=(DFX*XNUE+DFY*YNUE)/SQRT(DFX**2+DFY**2)
184      44 KK=3-K
185      41 DFI(KN,KK)=COSE
```

```
      COEF(NODO,KN)=COEF(NODO,KN)-B(K)*COSE
      CARGA(NODO)=CARGA(NODO)-A(K)*FI(KN)
      GO TO 80
    5 COSE=1
      GO TO 41
   80 CONTINUE
   90 CONTINUE
C
C  MODIFY THE ELEMENTS OF THE PRINCIPAL DIAGONAL IN THE COEFFICIENTS
C  MATRIX OF THE FUNCTIONS
C
      IF(NCOD(NODO)-1)91,92,91
   91 CARGA(NODO)=CARGA(NODO)+CDIAG*FI(NODO)
      GO TO 100
   92 COEF(NODO,NODO)=COEF(NODO,NODO)-CDIAG
  100 CONTINUE
C
C  SOLUTION OF THE SYSTEM OF EQUATIONS
C
      CALL SOLUG(COEF,NP,CARGA,IPIVO,NPMAX)
      DO 200 NODO=1,NP
      NC=NCOD(NODO)
C
C  ASSEMBLE AND MODIFY THE RESULTANTS VALUES OF THE SYSTEM OF EQUATIONS
C
      GO TO(201,202,202,204,205),NC
  201 FI(NODO)=CARGA(NODO)
      GO TO 200
  202 NC=NC-1
      DFI(NODO,NC)=CARGA(NODO)
      GO TO 200
  204 DO 210 I=1,2
C     IF(ABS(DFI(NODO,I)).LE=1.E-18) GO TO 210
      DFI(NODO,I)=CARGA(NODO)*DFI(NODO,I)
  210 CONTINUE
      GO TO 200
  205 DO 220 I=1,2
  220 DFI(NODO,I)=CARGA(NODO)
  200 CONTINUE
      CALL EXEC(8,NOM)
      END
C
C  THIRD SEGMENT STARTING
C
      PROGRAM BETI3(5)
      DIMENSION X(47),Y(47),NCOD(47),IENC(25),NOM(3),FI(47),DFI(47,2)
      DIMENSION XINT(46),YINT(46)
      DIMENSION FINT(46),A(2),B(2)
      COMMON IMP,LEC,NP,NPI,NSUP,IENC,NPMAX,X,Y,NCOD,FI,DFI,XINT,YINT
      DO 100 I=1,NPI
      FINT(I)=0.
      DO 50 J=1,NP
C
C  COMPUTE NUMERICALLY THE INTEGRATION VALUES ALONG BOUNDARY ELEMENTS
C
      CALL HGNUM(XINT(I),YINT(I),X(J),Y(J),X(J+1),Y(J+1),A(1),A(2),B(1),
     1B(2))
      JJ=J+1
      IF(J.EQ.NP)JJ=1
C
C  COMPUTE DE FUNCTION VALUE IN THE INTERNAL POINTS
C
   50 FINT(I)=FINT(I)-A(1)*FI(J)-A(2)*FI(JJ)+B(1)*DFI(J,2)+B(2)
     1DFI(JJ,1)
  100 FINT(I)=FINT(I)/(2*3.1415926)
      WRITE(IMP,2100)IENC
 2100 FORMAT(1H1,10X,25A2/10X,50("  ")/5X," BOUNDARY POINTS      "/6X,"NODE"
     1,11X," FUNCTION",6X."   FLUX BEFORE",5X,"      FLUX AFTER")
      WRITE(IMP,2101)(I,FI(I),(DFI(I,J),J=1,2),I=1,NP)
 2101 FORMAT(5X,I5,3F20.5)
      WRITE(IMP,2200)
 2200 FORMAT(1H1,5X,"INTERNAL POINTS"/4X"COOR X",4X,"COOR Y",6X," FUNCTI
     1ON")
      WRITE(IMP,2201)(XINT(I),YINT(I),FINT(I),I=1,NPI)
 2201 FORMAT(2F10.3,F15.3)
      STOP
      END
$
```